Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently Amended) A computing device, comprising:

a processor; and

memory having instructions stored therein, that are executable by the processor, to provide:

an application layer having executable instructions to provide one or more applications;

an operating system layer having executable instructions to provide a first type of operating system and associated application program interfaces (APIs), wherein the associated APIs include a first number of APIs for operating on the first type of operating system that is resident on the computing device; and

an interface module coupled between the application layer and the operating system layer, wherein the interface module includes a second number of APIs for operating with a second type of operating system that is not the type of operating system resident on the computing device and wherein the interface module receives program instructions from a program in the application layer written for the [[a]] second type of operating system and processes the instructions to select either, one of the first number of APIs or one of the second number of APIs ~~through emulation, interpretation, translation, and conversion by directing the instructions to APIs that correctly execute the instructions; and~~

~~wherein the interface module includes:~~

~~a discrete abstraction module having translation and conversion information therein; and~~

~~a discrete operating system emulation module in communication with the discrete abstraction module and having interpretation information therein.~~

2. (Original) The computing device of claim 1, wherein the interface module includes an operating system emulation module for emulating a number of operating system functions.

3. (Original) The computing device of claim 1, wherein the interface module emulates operating system functions and network server functions.

4. (Original) The computing device of claim 1, wherein the interface module emulates home location register functions.

5. (Original) The computing device of claim 1, wherein the interface module emulates intelligent network server functions.

6. (Original) The computing device of claim 1, wherein the interface module has portions for emulating the operating system functions and the network server functions in discrete modules located within the interface module.

7. (Original) The computing device of claim 1, wherein the interface module processes a program instruction by interpreting whether the instruction has to be processed further.

8. (Original) The computing device of claim 7, wherein the interface module converts a result received from the operating system layer such that the converted result is in a format that the application program can use to execute the instruction.

9. (Original) The computing device of claim 7, wherein the application interface module translates the instruction received such that the operating system layer can execute the instruction.

10. (Currently Amended) A system architecture, comprising:

a computing device including:

a processor; and

memory having instructions stored therein, that are executable by the processor, to provide:

an application layer having a home location register application thereon and having executable instructions to provide one or more applications;

an operating system layer having executable instructions to provide a first type of operating system and associated application program interfaces (APIs), wherein the associated APIs include a first number of APIs for operating on the first type of operating system that is resident on the computing device; and

an interface module coupled between the application layer and the operating system layer, wherein the interface module includes a second number of APIs for operating the home location register application with a second type of operating system that is not the type of operating system resident on the computing device and wherein the interface module receives program instructions from the home location register application in the application layer written for the second type of operating system and processes the instructions to select either one of the first number of APIs or one of the second number of APIs ~~to interface the home location register application designed for a second type of operating system with the first type of operating system through emulation, interpretation, translation, and conversion; and~~

~~wherein the interface module includes:~~

~~a discrete abstraction module having translation and conversion information therein; and~~

~~a discrete operating system emulation module in communication with the discrete abstraction module and having interpretation information therein;~~ and

a connection for connecting the computing device to a publicly switched telephone network (PSTN).

11. (Original) The system architecture of claim 10, wherein the interface module has a number of modules to translate instructions between the operating system layer and the application layer.

12. (Canceled)

13. (Canceled)

14. (Currently Amended) The system architecture of claim 10, wherein interface layer includes an [[the]] operating system emulation module that includes [[has]] translation and interpretation information therein.

15. (Original) The system architecture of claim 10, wherein the system architecture further includes an operating system emulation module to direct an instruction from the home location register application to an application program interface.

16. (Original) The system architecture of claim 10, wherein the system architecture further includes a number of component modules that can interface between an application designed for a second type of operating system and the operating system layer having a first type of operating system.

17. (Currently Amended) A method of executing an application comprising:

providing an application via an application layer having executable instructions to provide one or more applications to ~~configured for~~ an operating system layer having executable instructions to provide a first type of operating system and associated application program interfaces (APIs), wherein the associated APIs include a first number of APIs for operating on the first type of operating system that is resident on the computing device;

communicating instructions stored in memory and executable on a processor from the application to an interface module, wherein the interface module includes:

a second number of APIs for operating the home location register application with a second type of operating system that is not the type of operating system resident on the computing device and wherein the interface module receives program instructions from the application in the application layer written for the second type of operating system and processes the instructions to select either, one of the first number of APIs or one of the second number of APIs

~~a discrete abstraction module having translation and conversion information therein; and~~

~~a discrete operating system emulation module in communication with the discrete abstraction module and having interpretation information therein; and~~

~~processing the instructions with the interface module through emulation, interpretation, translation, and conversion to function with a different operating system~~.

18. (Original) The method of claim 17, wherein processing the instructions from the application with the interface module includes using a list of instructions to be processed.

19. (Original) The method of claim 17, wherein the application is configured for a Linux based operating system.

20. (Original) The method of claim 17, wherein the application is configured for a Windows based operating system.

21. (Original) The method of claim 17, wherein the application is configured for a UNIX based operating system.

22. (Original) The method of claim 17, wherein the method further includes identifying instructions to be translated by the interface module.

23. (Currently Amended) A method of executing an application configured for a platform having first type of operating system on a platform having a second type of operating system comprising:

communicating instructions from the application to an interface module, the application configured for a first type of operating system;

interpreting the instructions from the application with the interface module ~~through emulation, interpretation, translation, and conversion~~ by receiving program instructions from the application and processing the instructions to select either, one of a first number of APIs that are designed for use of the application on the first type of operating system and wherein the first number of APIs are resident on an operating system layer or one of a second number of APIs that are designed for use of the application on the second type of operating system and wherein the second number of APIs are resident on the interface module; and

communicating the instructions from the interface module to an operating system that is the second type of operating system~~; and~~

~~wherein the interface module includes:~~

~~a discrete abstraction module having translation and conversion information therein; and~~

~~a discrete operating system emulation module in communication with the discrete abstraction module and having interpretation information therein~~.

24. (Original) The method of claim 23, wherein communicating instructions from the application to an interface module includes communicating instructions to an operating system emulation module within the interface module.

25. (Original) The method of claim 24, wherein interpreting the instructions includes directing an instruction from the operating system emulation module to an application program interface.

26. (Original) The method of claim 23, wherein communicating instructions from the application to an interface module includes communicating instructions to a network server emulation module within the interface module.

27. (Original) The method of claim 23, wherein interpreting the instructions includes translating an instruction configured for the first type of operating system to an instruction configured for the second type of operating system.

28. (Original) The method of claim 23, wherein interpreting the instructions includes converting a result configured for the second type of operating system to a result configured for the first type of operating system.

29. (Currently Amended) A computer readable medium having a set of computer executable instructions thereon for causing a device to perform a method, comprising:

communicating instructions from a telecommunications application to an interface module, the telecommunication application configured for a first type of operating system;

processing the instructions from the telecommunication application with the interface module by receiving program instructions from the application and processing the instructions to select either, one of a first number of APIs that are

designed for use of the application on the first type of operating system and wherein the first number of APIs are resident on an operating system layer or one of a second number of APIs that are designed for use of the application on a second type of operating system and wherein the second number of APIs are resident on the interface module ~~through emulation, interpretation, translation, and conversion~~; and

communicating the instructions from the interface module to an operating system that is a second type of operating system;

~~wherein the interface module includes:~~

~~a discrete abstraction module having translation and conversion information therein; and~~

~~a discrete operating system emulation module in communication with the discrete abstraction module and having interpretation information therein~~.

30. (Original) The computer readable medium of claim 29, wherein communicating instructions from an application to an interface module includes communicating to an abstraction module within the interface module.

31. (Original) The computer readable medium of claim 29, wherein communicating instructions from an application to an interface module includes communicating instructions to a component module within the interface module.

32. (Original) The computer readable medium of claim 29, wherein the method further includes identifying instructions to be converted by the interface module.